Q1. Select three software architecture patterns from the ten discussed in https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013 and explain their features, advantages and disadvantages.

1. Client-server pattern

The client-server pattern consists of two parties, as the name implies, a server and clients. The server component will provide services to multiple client components while clients request servers from the server and the server provides relevant services to those clients. In addition, the server continues to listen to client requests. One primary advantage of this pattern is that servers can be distributed across a network meaning that general functionality can be available to all clients and does not need to be implemented by all services. There is one principal disadvantage of the client-pattern pattern in which each server is a single point of failure so it makes this system susceptible to denial of service attacks or server failure. Performance may be unpredictable as well because it depends on the network as well as the system.

2. Pipe-filter pattern

In the pipe-filter pattern, this pattern can be used to structure systems which produce and process a stream of data. Within the processing of data, a filter component is involved in which each processing step is enclosed within a filter component. In the processing, data that is to be processed is passed through pipes. Advantages of the pipe-filter pattern include being easy to understand and supporting transformation reuse, evolution by adding transformations is straightforward, and it can be implemented as either a sequential or concurrent system. Disadvantages of this pattern involves the agreement of the format for data transfer between communicating transformations.

3. Model-view-controller pattern

The features of the model-view-controller pattern include dividing an interactive application into 3 parts, the model component in which it maintains the core functionality and system data, the view component where it manages how the data is displayed to the user, and the controller where it handles the input from the user such as key presses or mouse clicks. This pattern allows us to separate presentation and interaction from the system data. Advantages of this pattern include allowing the data to change independently of its representation and vice versa. It also supports presentation of the same data in different ways with changes made in one representation shown in all of them. Disadvantages include the possibility of additional code and code complexity when the data model and interactions are simple.